

# **LCInvert**

## **Installation**

**and**

## **Operating Instructions**

Bdw Publishing  
17995 Bakers Farm Rd.  
Colorado Springs, CO 80908

<http://www.MinorPlanetObserver.com>

© 2009 All Rights Reserved

# Table of Contents

---

<b>LICENSE AGREEMENT FOR SOURCE CODE .....</b>	<b>5</b>
THE C AND DELPHI CORE LIBRARY SOURCE CODE.....	5
<b>OPENGL REQUIREMENTS .....</b>	<b>7</b>
<i>Video Drivers</i> .....	7
<i>Video Card Settings</i> .....	7
<b>INSTALLATION .....</b>	<b>9</b>
<b>SHAPE AND SPIN AXIS MODELING PRIMER.....</b>	<b>13</b>
PROCEED WITH CAUTION! .....	13
HOW MUCH IS ENOUGH DATA? .....	14
RELATIVE VERSUS ABSOLUTE DATA.....	15
SOME MORE TECHNICAL NOTES .....	16
<i>More on Relative versus Absolute</i> .....	16
<i>Convex Hulls</i> .....	16
<i>Regularization Weight and Albedo Variation</i> .....	16
THE BOTTOM LINE .....	17
<b>THE CORE LIBRARIES .....</b>	<b>19</b>
<i>Support/Documentation</i> .....	19
<i>C Code</i> .....	19
<i>C Code Manual</i> .....	19
<i>Delphi Code</i> .....	23
<b>LCINVERT OVERVIEW .....</b>	<b>25</b>
MAIN MENU .....	26
<i>File Menu</i> .....	26
<i>Utilities Menu</i> .....	26
<i>Windows Menu</i> .....	27
ABOUT THE TUTORIALS .....	27
<b>WORKING WITH MODEL DATA.....</b>	<b>29</b>

WHAT IS MODEL DATA? .....	29
RELATIVE SIZES OF MODELS.....	29
LOADING MODEL DATA.....	30
OPENING A VIEW .....	31
<i>The Model View</i> .....	32
<i>Arbitrary Date</i> .....	36
<i>The 4-Vane View</i> .....	37
<b>TUTORIALS .....</b>	<b>39</b>
<b>DATA CONVERSIONS .....</b>	<b>41</b>
MPCORB CONVERT .....	41
STEPS TO CONVERT THE FILES .....	42

# License Agreement for Source Code

---

This manual must be included with the any distribution of the lightcurve inversion source code and/or executable program. While it contains documentation specific to the commercial product, LCINVERT, it's hoped that documentation will provide guidance to those writing their own programs on how to make effective use of the core library code.

## The C and Delphi Core Library Source Code

The C and Delphi source code files for the core libraries are distributed royalty free and provided "as-is". See below. The source code is provided to allow programmers to construct their own programs using the original code and, if they are inclined, to try to improve upon the original code and distribute it. As noted below, any distribution, be it the core libraries and/or an executable program must always include the original code and documentation.

### Standard File Header and Credit

The following must be included at the top of any source file, whether it is the original source code, a modification, or additional source code distributed with the original source code. Note that the CREDIT section must appear automatically when running an executable file and be available within the program, e.g., in the About Box.

#### DISTRIBUTION/DISCLAIMER NOTICE

THIS DISTRIBUTION NOTICE AND DISCLAIMER MUST BE INCLUDED WITH ANY DISTRIBUTION OF THE LIGHTCURVE INVERSION CORE LIBRARY CODE.

THIS SOURCE CODE IS PROVIDED "AS-IS" WITHOUT ANY CLAIMS FOR SUITABILITY OR ACCURACY. IT IS PROVIDED WITHOUT ANY WARRANTY, EXPLICIT OR IMPLICIT, OR WITH ANY LIABILITY. THE USER IS SOLELY RESPONSIBLE FOR ITS USE AND SUBSEQUENT CONSEQUENCES.

#### TERMS OF USE

-----

THE USER IS PROVIDED THIS CODE ROYALTY FREE.

THE SOURCE CODE MAY BE USED IN A PROGRAM FOR PERSONAL, FREeware, SHAREWARE, AND/OR COMMERCIAL USE.

THE USER MAY MODIFY THE SOURCE CODE, BUT NOT THE ORIGINAL DOCUMENTATION, AS HE WANTS, PROVIDING:

1. ALL ORIGINAL FILES, INCLUDING SOURCE CODE AND DOCUMENTATION, ARE DISTRIBUTED WITH ANY LIBRARY, FREeware, SHAREWARE, AND/OR COMMERCIAL PRODUCT DISTRIBUTION. THIS INCLUDES THE LCINVERT.PDF DOCUMENTATION.
2. THIS DISCLAIMER NOTICE IS INCLUDED AT THE TOP OF EVERY SOURCE CODE FILE.
3. PROPER CREDIT TO THE ORIGINAL AUTHORS AND THE JOURNAL REFERENCES IS GIVEN IN ALL ADDITIONAL DOCUMENTATION. THIS INCLUDES ANY POPUP HELP, SPLASH, OR ABOUT SCREENS OF A FREeware, SHAREWARE, AND/OR COMMERCIAL EXECUTABLE PROGRAM.
4. THE CREDIT STATEMENT MUST APPEAR AUTOMATICALLY AT START UP, E.G., A SPLASH SCREEN IN ANY FREeware, SHAREWARE, AND/OR COMMERCIAL PRODUCT. A PERSONAL PROGRAM IS EXEMPT PROVIDING IT IS USED ONLY BY THE USER AND IS NOT GIVEN TO ANY OTHER PERSON.

ANY PROGRAM USED BY SOMEONE OTHER THAN THE PROGRAMMER IS CONSIDERED TO FALL UNDER THE CATEGORY OF EITHER FREeware, SHAREWARE, AND/OR COMMERCIAL AND MUST DISPLAY THE CREDITS AUTOMATICALLY.

CREDITS

-----

THE FOLLOWING CREDIT STATEMENT SHALL BE INCLUDED IN ALL ADDITIONAL DOCUMENTATION AND ANY OF THE FOLLOWING: HELP, SPLASH, OR ABOUT SCREENS FOR FREWARE, SHAREWARE, AND/OR COMMERCIAL PRODUCTS.

This program is based on original FORTRAN source code by Mikko Kaasalainen.  
The C source code conversion was written by Josef Durech.  
The Delphi source code conversion was written by Brian D. Warner.

REFERENCES

Kaasalainen, J., Torppa, J. "Optimization Methods for Asteroid Lightcurve Inversion: I. Shape Determination". 2001. Icarus 153, 24-36.

Kaasalainen, J., Torppa, J., Muinonen, K., "Optimization Methods for Asteroid Lightcurve Inversion: II. The Complete Inverse Problem". 2001. Icarus 153, 37-51.

Kaasalainen, M., Mottola, S. Fulchignoni, M., "Asteroid Models from Disk-integrated Data" in Asteroids III. 2002, 139-150.

Kaasalainen, M., Durech, J., "Inverse Problems of NEO Photometry: Imaging the NEO Population", Proceedings IAU Symposium No. 236, in press.

<http://www.rni.helsinki.fi/~mjk/asteroids.html>  
(see the FAQ and the numerous references)

# OpenGL Requirements

The LCInvert program uses Delphi 7 components that use OpenGL for 3-D rendering. Most Windows versions come with the OpenGL DLL (Win98 and above). This alone does not guarantee that running the LCInvert program will properly display the 3-D models.

There is no setting in the LCInvert program or components that affects whether or not your video card will properly display the 3-D models. As was found during beta testing, all display problems were eventually fixed by updating drivers and/or video settings.

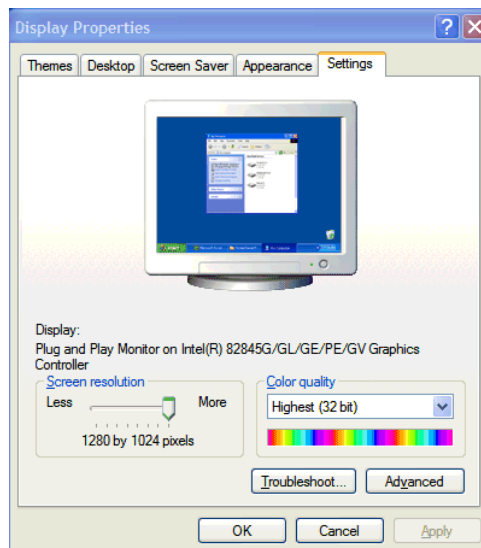
## Video Drivers

The most common cause when running LCInvert and not seeing the model properly displayed has to do with the video drivers for your graphics card. During beta testing, several users had problems and almost all had nVidia graphics cards. A visit to the nVidia web site found a 40MB download that, once installed, allowed LCInvert to run properly. If you have trouble running LCInvert, please check the web site of the – graphics card manufacturer for updated drivers and/or information about running OpenGL.

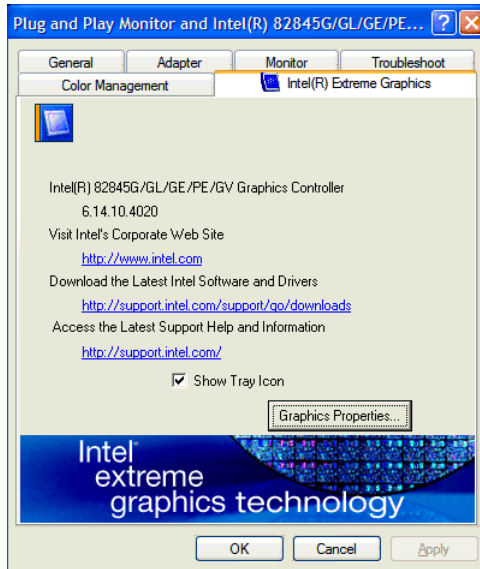
## Video Card Settings

While you may have the proper drivers, the video card settings may not be correct. The following steps may help in that they show the setup for one particular brand of card. The screen shots may be different for your system and there may not be a tab for OpenGL at all. Again, consult with the video card manufacturer.

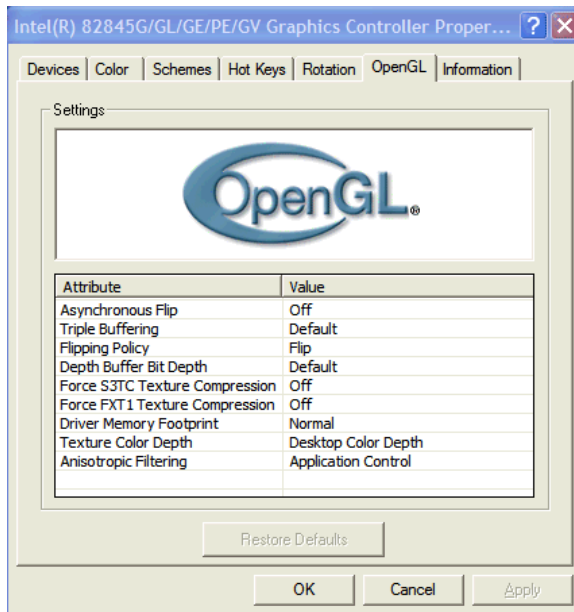
1. Right click on the Windows desktop and select "Properties" from the popup menu.
2. Click on the Settings tab.



3. Click on the Advanced button. This will display a another form. See if there is a tab specific to your video card. If so, click on that tab.



4. In the example above, there is a button labeled Graphics Properties. There may be something similar to what you see on your machine. If so click it to display the next form.



5. For this particular video card, there is a tab dedicated to OpenGL. If you see something similar (it may be a tab or it may be a button that opens another form), click the tab or button. Check the manual for the card, or visit the manufacturer's web site, for information about the settings. See if changing one or more of the settings, which might include turning support on or off, helps resolve the problems viewing 3-D models in LCInvert.



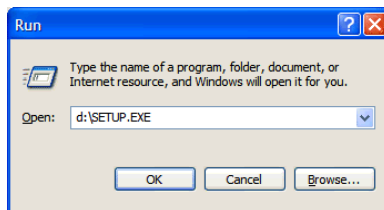
# Installation

The LCInvert program installation is very similar to that for the full MPO Software Suite. Only those files required for LCInvert are part of the installation package but those files that are common to the MPO Suite installation are placed in the same directories in order to avoid duplicate files and confused registry entries.

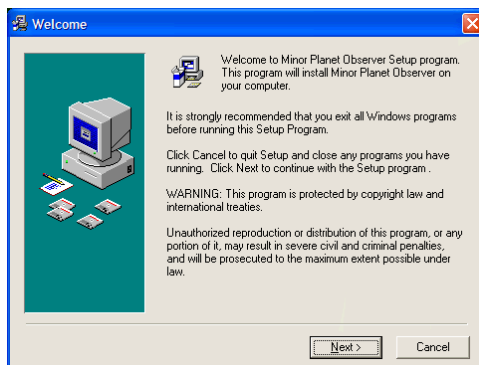
↪ Be sure to read the instructions in this section as you proceed. Otherwise, problems could result if you are installing on the same machine where other MPO software has already been installed.

↪ Close ALL MPO programs before starting the installation.

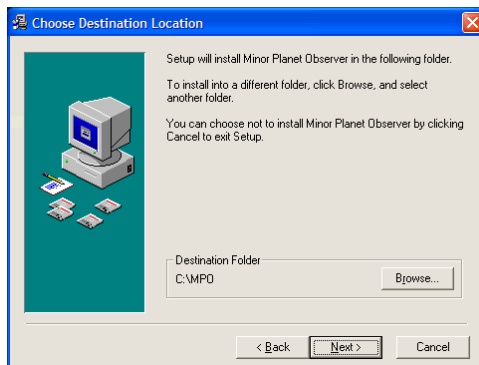
1. Insert the DVD into the DVD drive.
2. Select Start | Run on the Windows menu.



3. Enter X:\SETUP.EXE in the Open field, where X is the CD (or DVD) drive. After a few moments, the LCInvert installation program starts.



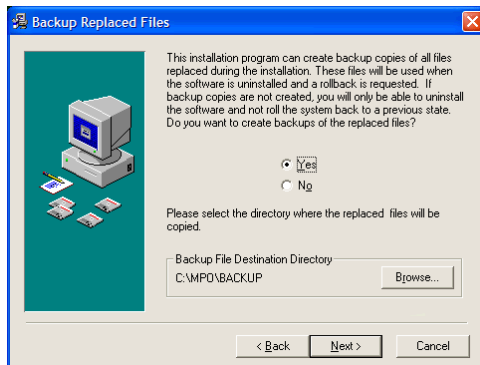
4. Click Next to go to the next page.



5. Click Browse to change the destination folder.

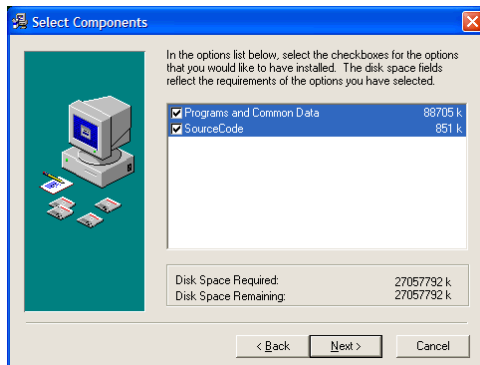
➤ If you are installing on a machine where other MPO software has been installed, you must set the destination folder to the base directory for the other MPO software, usually C:\MPO.

6. Click Next to proceed to the next page.



This step is optional and is not usually recommended.

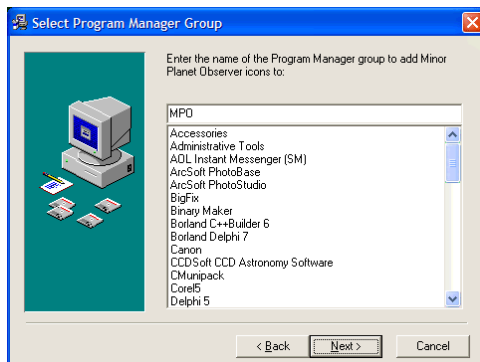
7. Click Next to proceed to the next page.




8. Select the items to be installed. If you select "Programs and Common Files", you must have a release code (password) to complete the installation. This is provided only to those who purchased LCInvert.

➤ Regardless of your selection, the sample data is always installed into <BASEDIRECTORY>\DATA and <BASEDIRECTORY>\DATA\BACKUP. The latter is a duplicate copy of the files in case you want to restore the sample files to their original state without reinstalling.

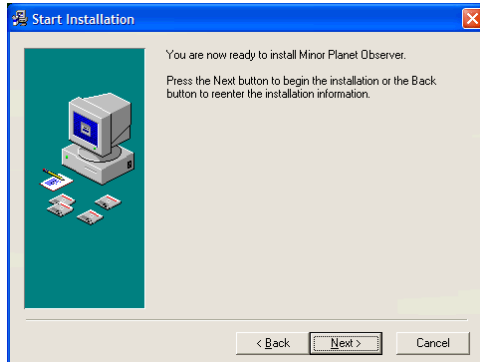
9. Click Next to proceed to the next page.



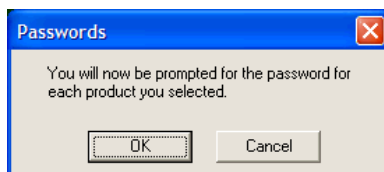
10. Change the Group name if you want.

 If you are installing on a machine where other MPO software has been installed, use the same group name as was used for the other MPO software, usually MPO.

11. Click Next to proceed to the next page.



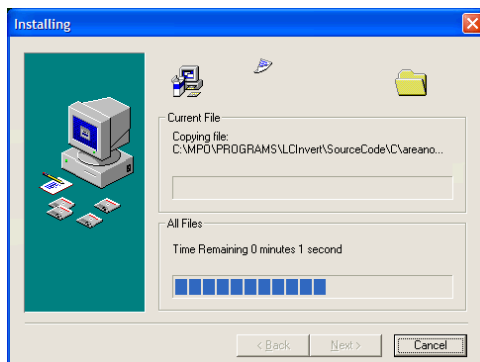
12. Click Next to proceed to the next page. If you selected to install the LCInvert program, a special message is displayed. If you are installing only the source code, skip to step 15.



13. Click OK to proceed.

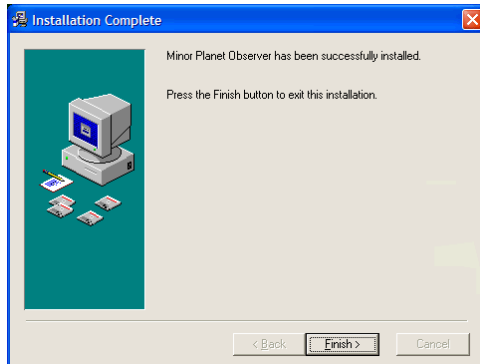


14. Enter the release code on the card that came with your CD or DVD. Then click OK to start the installation.



15. The installation program copies the files to your computer.

16. When the copying is done, click Next to proceed to the next page.



17. Click Finish to close the installation program.

The installation places an icon on the Windows desktop as well as creating a program group within the Start | Programs menu.

### The MPO Server

All MPO programs are built on the concept of "client-server". The programs are the clients, which make requests for data to a "server" program. The installation program installs this server program and automatically configures it for you. You should not have to make any changes to the MPO Server.

The server program is included in the Startup group so that it is running after Windows boots. However, just in case something went wrong, all MPO programs check for the presence of the server and, if it is not running, automatically start the server for you.

## Shape and Spin Axis Modeling Primer

---

In 1906, no less than Henry Norris Russell declared that it was impossible to determine the shape of an asteroid from its lightcurve. His argument ran that an asteroid could be "painted" (albedo variations) such that even when the maximum area was presented to an observer, the lightcurve would be at a minimum. This theory held forth for nearly 75 years, until new methods were developed, methods that presumed – as is now known to be the case – that asteroids are generally uniformly gray. There are albedo variations but they contribute only a small fraction of the total amplitude change as an asteroid rotates.

Once it was accepted that shape of the asteroid, not albedo, was the dominant factor in the shape and amplitude of a lightcurve, the problem of lightcurve inversion was no longer insurmountable. Several methods were developed that could at least produce a spin axis solution, i.e., the ecliptic longitude and latitude to where the north pole of the asteroid was pointing along with sidereal period. However, these were not highly reliable nor did they produce good shape models – 3-D renderings of the asteroid. You are referred to Clifford Cunningham's *Introduction to Asteroids* and several chapters in *Asteroids II* for additional background on the early methods of shape and spin axis modeling.

A significant break through occurred in the early years of the 21<sup>st</sup> century with the Mikko Kaasalainen et al and others. To be honest, the math is a bit deep and well beyond the scope of this manual, but you should make a point of reading at least three publications and one web site:

Kaasalainen, J., Torppa, J. "Optimization Methods for Asteroid Lightcurve Inversion: I. Shape Determination". 2001. *Icarus* **153**, 24-36.

Kaasalainen, J., Torppa, J., Muinonen, K., "Optimization Methods for Asteroid Lightcurve Inversion: II. The Complete Inverse Problem". 2001. *Icarus* **153**, 37-51.

Kaasalainen, M., Mottola, S. Fulchignoni, M., "Asteroid Models from Disk-integrated Data" in *Asteroids III*. 2002, 139-150.

Kaasalainen, M., Durech, J., "Inverse Problems of NEO Photometry: Imaging the NEO Population", 2007. Proceedings IAU Symposium 236, in press.

<http://www.rni.helsinki.fi/~mjk/asteroids.html> (see the FAQ and the numerous references)

### Proceed with Caution!

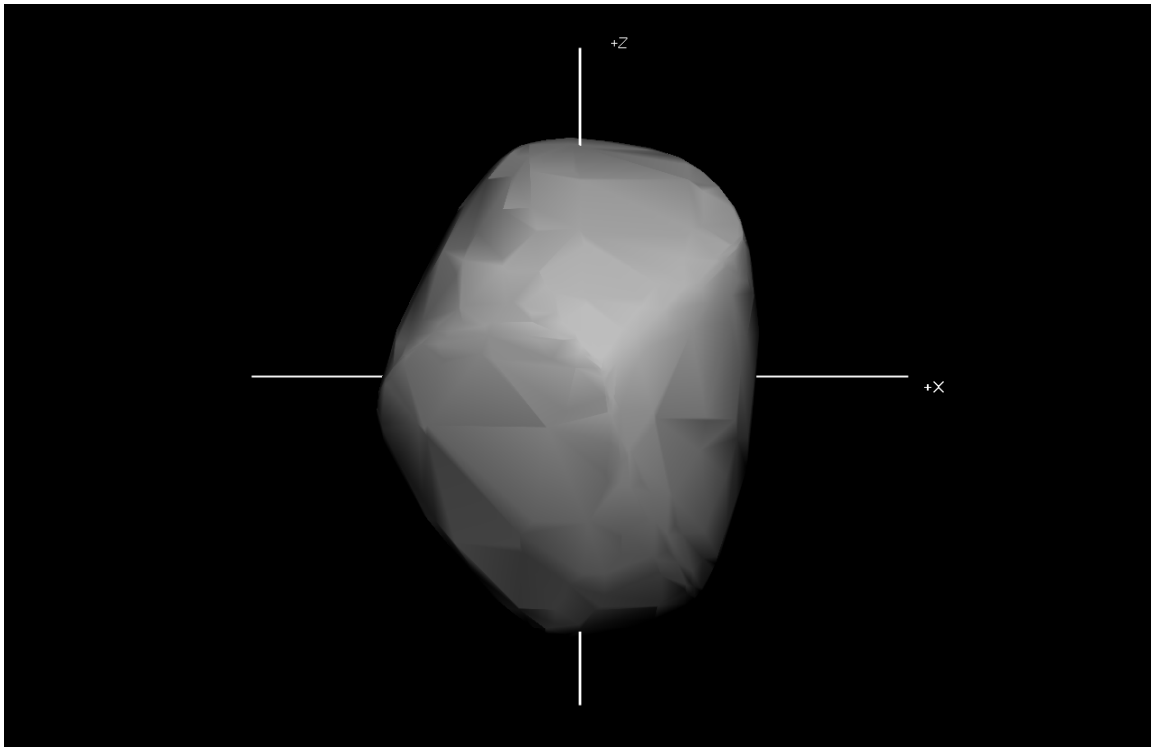
It is very important to emphasize that lightcurve inversion is not a simple, straightforward process and is fraught with many more pitfalls than might ever be encountered when finding the period of a lightcurve. ***It is very easy to reach false conclusions.*** A certain amount of data is required that meets specific requirements.

To answer the most common question right off: ***You cannot get the shape or spin axis of a main-belt asteroid from data during a single apparition. It just cannot be done.***

An NEO that goes through a significant change in phase angle and viewing aspect during a single apparition might be an exception, but only if there is sufficient data.

When doing lightcurve inversions, it is highly recommended that you contact those in the literature that have done it successfully and get their feedback ***before publishing any work.***

For example, the model in the screen shot below was the result of doing a search and assuming the model with the best fit, lowest ChiSq value,(see below), was a valid solution. In this case, all the ChiSq values were very similar, which is an almost sure indication that the available data cannot generate a good model.



Any model where its longest axis is the axis of rotation should cause immediate concern!

## How Much is Enough Data?

The best source is Kaasalainen, who wrote the following (with some paraphrasing) in the first two articles mentioned above. The added emphasis is for this manual and not in the original manuscript.

### Relative versus Absolute Data

Test simulations indicate that the solution based on relative brightness is, for all practical purposes, equivalent to that obtained from absolute photometry [*the two are explained below – Bdw*]. This is quite understandable: it is primarily the shapes of lightcurves that are strongly connected with the pole, the period, and the shape of the asteroid.

### Lightcurve Data Requirements

Not very many input lightcurves are needed for inversion. They suffice *so long as they cover a wide range of observing geometries* and they are sufficiently many lightcurve points. The main requirement for the data sets is that they consist of lightcurves that exhibit "characteristic" features caused by the shape.

In practice this means the observing geometries must reach large solar phase angles ( $a$ ); due to the lack of shadowing effects, shape information in observation made at small solar phases is often restricted to the general dimensions of the target. For a more detailed solution, there should be at least a few lightcurves with  $a$  greater than, say,  $20^\circ$ .

The density of lightcurve points in rotational phase does not seem to be crucial as such: if the observing geometries produce characteristic lightcurves, it does not necessarily matter if there are 20 curves with 50 points each, or 10 curves with 100 points each.

Even very complex lightcurves can be modeled easily and consistently using convex inversion. If the data sets cover sufficiently many geometries, the resulting pole, period, and shape solutions are stable regardless of noise or the scattering law(s) used. The potential insuffi-

ciency of the data shows during the inversion procedure; *even large data sets may not be as informative as smaller but more varied ones.*

Some lightcurves contain hundreds of point, while most have 30-70 and some only a few; also, some dates may be represent more heavily than others. This is a potential source of biasing effects; however, visual checking of plots shows whether some lightcurves have been unduly favored at the expense of others, and separate weighting factors (also taking into account the quality of the lightcurves) can be applied. So far, we have not found any such factors necessary, but it is sometimes expedient simply to remove redundant or bad lightcurves from the data set.

To put it in a nutshell, pole, period and the photometric convex hull of the object overwhelmingly dominate lightcurve morphology. Concavities and scatter properties are, in a way, comparable to noise: even though changing them does change the lightcurves somewhat in the direct problem, the solution of the inverse problem does not change much since there are no notably different convex shapes that could model the lightcurves better. The well-posedness and stability of the inversion procedure is thus a result of the convex formulation.

To summarize, for a good shape and spin axis solution:

1. Lightcurves of sufficiently different geometries are required. (Which is why you can't get a good solution for a main-belt asteroid from one apparition since the viewing geometry is near the same for the whole apparition).
2. Lightcurves within the data set should cover a range of phase angles, including some at phase angles of  $>10^\circ$  and, for an even better solution,  $>20^\circ$ .
3. The range of geometries and phase angle coverage is more important than the density of the data set, though excessively sparse lightcurves may be a hindrance.
4. Removing nearly redundant curves (in date) or bad (too sparse, high noise) can help remove biases and is more expedient than trying to apply weighting to individual curves.
5. The synodic period must be known to a sufficient degree of precision. The smallest separation  $\Delta P$  of local minima in the period parameter space is roughly given by

$$\Delta P \approx 0.5 \frac{P^2}{\Delta t}$$

where  $\Delta t$  is the full epoch range of the data set. This derives from the fact that the maxima and minima of a double-sinusoidal lightcurve for periods  $P$  and  $P \pm \Delta P$  are at the same epochs after  $\Delta t$  time.



The period and time span must be in the same units. The result is in those same units.

For a good solution, the initial period in the search should differ from the true sidereal period by no more than the  $\Delta P$  given in the formula above. For example, assume the period is 6.0000 hr (0.25000 d) and that the observations span a range of five years, or about 1800 days. In this case,  $\Delta P$  is about 0.0004 hr. If the span of data covers 30 years, about 11,000 days, then  $\Delta P$  is about 0.000007 hr. If these levels of precision are not possible, then you should use the period scan available in LCInvert's Inversion Wizard, which is based on the period\_scan C source code.

## Relative versus Absolute Data

Absolute (or calibrated) data means that all the magnitudes (or intensities) have been put on a common system. This system can be internal or it can be one of the standard magnitude bands such as Johnson V. In addition, the data should be corrected to unity distance by applying, for magnitudes,

$$-5 \log(\Delta r)$$

Where  $\Delta$  is the asteroid-Earth distance and  $r$  is the asteroid-Sun distance, in AU.

If data in different lightcurves are absolute (calibrated), they should not be shifted by arbitrary amounts within themselves to get the different curves to match. Supposedly, by virtue of being calibrated, they already do so.

In fact, you can take so-called "absolute data" (many times it is said to be but really isn't) and back out the unity distance correction if it was applied. The data set *can then be used as a relative data set*. This is done quite frequently since it is **strongly** recommended that only relative data sets be used when doing modeling.

Calibration is needed only for the sparse data sets anticipated from the Pan-STARRS and other surveys since the time span of the observations will be years and so the data needs to be measured on a consistent zero point. The same might be said if following an asteroid with a long period, i.e., days, since small sections of the curve are nearly flat and so fitting them together properly is difficult without calibration. Again, once the entire data set is calibrated within itself, it is usually treated as a single relative set.

As you might gather from the above, a relative set of data has not formally been put on an internal or standard system. As often occurs with amateur (and even some professional work), data from several different nights are arbitrarily shifted to get a best fit. ***This is still relative data***, by virtue that the shifts are arbitrary. So, in general, amateurs are already providing the necessary lightcurve data required for modeling.

The ability to use relative data is what makes amateur work important to and easy to incorporate within the planned efforts to use so-called sparse data sets generated by the large surveys to produce possibly thousands of shape and spin axis models in the first decade of the Pan-STARRS survey. In short, the sparse data sets will generate good solutions in almost all cases. However, for refined solutions and to resolve unusual case (possible binary, etc.), dense lightcurves will be required. This is where the amateur community will fill in the gaps – by supplying just a few dense lightcurves of a given object. Of course, Pan-STARRS will reach much fainter than most amateurs, but there will still be more than enough work to occupy dozens observers for many years.

## Some More Technical Notes

This section provides a little more technical background on modeling.

### More on Relative versus Absolute

When using relative data set the z-axis is poorly constrained. You can change the z-axis dimension and the relative curves change very little if at all.

Where absolute data becomes important is when the z-axis must be properly constrained, e.g., "hamburger-shaped" asteroids since there is very little info in the lightcurve shape to start. The absolute data allows getting the proper level of flattening.

### Convex Hulls

The modeling code provided with the distribution and used in LCInvert presumes that only convex shapes form the entire model. In reality, this is not true, e.g., Eros with its saddle and Mathilde with its huge craters. What the modeling code does is create a shape as if one used wrapping paper to go around the actual object. Where there are concavities, the wrapping paper will form a flat area. Indeed, any model with a large flat area most likely represents a large crater or saddle feature.

This is discussed in more detail in the papers given in the references section, where it is shown that assuming convex shapes produces nearly identical models as when assuming otherwise and requires considerably less complex treatments and more consistent results.

### Regularization Weight and Albedo Variation

One of the input parameters for the model search is a weighting factor given to the "dark facet". This facet is added as an absolute lightcurve (regardless of the data being used) that forces the model to be convex. *It has no effect on the lightcurve of the resulting model.*



The initial modeling process generates a file that includes the area and xyz coordinates of the normal to the area. For a good result, the area of the dark facet must be less than 1% of the total area. The OUT\_PAR files generated by LCInvert include that percentage. If the area is too large, increasing the weight factor should decrease the size of the dark facet. This is particularly true if the z-component of the normal of the dark facet is large, i.e., the normal is generally pointing up or down.

However, if the dark facet area remains large with increased weighting and its normal is not pointing close to up or down, i.e., more parallel to the equator, this means there very likely a real albedo feature on the asteroid such as a particularly bright or dark spot. This is very rare but does happen. In this case, increasing the weight doesn't fix the area problem and worsens the fit.

In this regard, use the ChiSq value of the fit for the model in comparison to that from other models to find the best model.

## The Bottom Line

When can you say that you have a solution that is likely correct? The manual for the C code (starting pg. 19) gives some answers. Put succinctly, if your solution does not stand out from the set of many trials (initial periods and poles, at least at a level of about 10% of the chi-square), or it does not fit (most) data to noise level, then you do NOT have a good solution.

↪ The period\_scan C code (see pg. 22) or the AutoSearch method in LCInvert find multiple solutions and allow you to review them in light of the above standards.



## The Core Libraries

---

The LCInvert and Libraries distributions both come with the C and Delphi 7 core library source code. The source code for LCInvert program is not included. In addition, a Borland C++ 6 project (BPR) and its executable are included so that those using that compiler can quickly create a running project based on the C code. The EXE is a "console application".

### Support/Documentation

The source code is considered to be its own documentation. It is the programmer's responsibility to understand the code and how it works.

### C Code

The C code was written by Josef Durech and is based on the original FORTRAN code written by Mikko Kaasalainen. The following files comprise the C code library:

areanorm.c	gauss.c	sphfunc.c
blmatrix.c	lubksb.c	trifac.c
bright.c	ludcmp.c	
conv.c	matrix.c	constants.h
convexinv.c	memory.c	delcarations.h
covsrt.c	mrqcof.c	globals.h
curv.c	mrqmin.c	
dot_product.c	period_scan.c	minkowski.f
ellfit.c	phasec.c	standardtri.f

↳ Note the two FORTRAN files, Minkowski and StandardTri. These are not available in C at this time. Users must convert those to C themselves. The FORTRAN code is readily deciphered and so the conversion is not very difficult.

### C Code Manual

The following was prepared by Josef Durech for distribution with the C code library.

↳ While some of the following is specific to the C code input and output files, there are important points made about the initial search parameters and the overall inversion process.

The ConvexInv program computes the shape+spin+scattering model that gives the best fit to the input lightcurves (calibrated, uncalibrated, or sparse). Scattering law+shape representation are simple and robustly converging. The shape representation this procedure obtains is the Gaussian image of a convex polyhedron, i.e., the areas of the facets (with fixed outward normals).

The vertices are directly solved by the Minkowski procedure minkowski.

Syntax (Unix):

```
cat lcs | convexinv [-v] [-s] [-o out_areas] [-p out_par] input_par out_lcs
```

-v verbose mode

-s puts output areas to standard output; do not use it together with -v

When run with the -v option, the program writes down values of  $\chi_{\text{rel}}^2$  rms deviation  $\sqrt{\chi_{\text{rel}}^2/N}$  (where N is the number of data, and  $\lambda$  parameter of the Levenberg-Marquardt algorithm. The relative chi-square is defined as

$$\chi_{\text{rel}}^2 = \sum_i \left\| \frac{L_{\text{obs}}^{(i)}}{\bar{L}_{\text{obs}}^{(i)}} - \frac{L^{(i)}}{\bar{L}^{(i)}} \right\|^2,$$

where  $L_{\text{obs}}^{(i)}$  and  $L^{(i)}$  are observed and modeled lightcurves and they are renormalized through the average brightness  $\bar{L}_{\text{obs}}^{(i)}$  and  $\bar{L}^{(i)}$ .

At the end, the solution of the pole, period, and scattering parameters is written, together with the dark facet area (see below).

### Input lightcurves (lcs)

The input file contains lightcurve data and the corresponding geometry, it is read from the standard input. The first line gives the total number of lightcurves, then the individual lightcurves follow in 'blocks'. Each lightcurve starts with the number of points and 0/1 code for a relative (0) or calibrated (1) lightcurve. Then there are lines with the epoch in JD (light-time corrected!), the brightness in intensity units (reduced to unit distances from the Earth and the Sun when calibrated), the ecliptic *astrocentric* cartesian coordinates x, y, z of the Sun and of the Earth in AU.

For a slowly moving main belt asteroid, the coordinate vectors can be approximated as to be constant for a single-night lightcurve.

### Input Parameters (input\_par)

*Initial spin* – asteroid's initial ecliptic pole coordinates  $\lambda$ ,  $\beta$  (deg), and the rotation period P (hours). The initial values are followed by 0 or 1, depending on whether those are fixed (0) or free (1).

*Zero time  $t_0$  (JD), initial rotation angle  $\phi_0$  (deg)* – they are needed for the transformation between vectors  $r_{\text{ast}}$  in the asteroid co-rotating coordinate frame and vectors  $r_{\text{ecl}}$  in the ecliptic coordinate frame. The transformation is given by the equation

$$r_{\text{ecl}} = R_z(\lambda)R_y(90^\circ - \beta)R_z\left(\phi_0 + \frac{2\pi}{P}(t - t_0)\right)r_{\text{ast}},$$

where  $R_i(\theta)$  is the rotation matrix corresponding to the rotation angle  $\theta$  along the  $i$ -axis in the anti-clockwise direction,  $t$  is the time, and P is the rotation period.

If  $t_0 \leq 0$  in the input file, then it is set to the lowest JD epoch of the data set. So the best choice is  $t_0 = 0$  and  $\theta = 0$ .

*Convexity regularization weight* – this is often needed to keep the shape formally convex. A typical value is 0.1, but it may often have to be orders of magnitude larger or smaller. There is a dark facet area that makes the whole set of facets convex. Always try to put it below 1% of the total area by increasing the convexity regularization parameter. If you cannot get a good fit with a small dark area and if the fit significantly improves when the dark facet is large, it means that there is likely to be an albedo variation over the surface. See "Regularization Weight and Albedo Variation" on pg. 16 for more information.

*Laplace series expansion – degree l, and order m.* This affects the number of shape parameters. If  $l = m$ , the number of shape parameters is  $(l + 1)^2$  (and  $((l + 1)^2 - 1)$  for relative lightcurves). A good choice is  $l = m = 6$ .

*Resolution* – number  $n$  of triangulation rows per octant (typically 8-10). The number of surface areas of the Gaussian image is  $8n^2$

*Light scattering parameters* – amplitude  $a$ , width  $d$ , slope  $k$ , and Lambert's coefficient  $c$ . Initial values are followed by 0/1 code for fixed (0) or free (1) parameters. The exponential-linear formulation of the phase function in the form

$$f(\alpha) = a \exp\left(-\frac{\alpha}{d}\right) + k\alpha + 1$$

is used in the code. The complete scattering model is

$$S(\mu, \mu_0, \alpha) = f(\alpha)[S_{LS}(\mu, \mu_0) + cS_L(\mu, \mu_0)],$$

where  $\alpha$  is the solar phase angle,  $\mu_0$ ,  $\mu$  are cosines of angles of incidence and reflection respectively,  $S_L$  is Lambert's law

$$S_L = \mu_0 \mu$$

and  $S_{LS}$  is the Lommel-Seeliger law

$$S_{LS} = S_L / (\mu + \mu_0).$$

Start with, e.g.,  $\alpha = 0.5$ ,  $d = 0.1$ ,  $k = -0.5$ . If you use only relative lightcurves, no solar phase function is needed, i.e., relative fit with almost constant solar phase for each lightcurve, then use fixed values of phase function parameters with zero amplitude and slope. Fitting the phase parameters using relative data leads to divergence. The parameter has usually only little effect on the solution so you can fix it at, for example,  $c = -0.1$ .

*Iteration stop condition* – if this is an integer number higher than 1, then it is the number of iteration steps in the Levenberg-Marquardt loop. If it is lower than 1, then it is the smallest difference in rms deviation between two subsequent steps. When the steps have a smaller difference, the iteration stops.

### **Output lightcurves (out\_lcs)**

In this file the brightness (in intensity units) of the model is stored. The file contains a list of brightness values in the same order as in the input lightcurves file. One can easily check the fit visually by plotting the observed lightcurves together with the modeled brightness.

### **Output areas and vertices (out\_areas)**

The first line gives the number of facets, then follow facet areas and outward unit normal  $x$ ,  $y$ ,  $z$  coordinates. This is used as an input for the Minkowski procedure that creates 3D model (see below). Note that the program writes the size and normal coordinates of the dark facet at the end, so the number of facets is  $8n^2 + 1$ .

The dark facet should be very small (controlled with the convexity regularization parameter). It is needed to make the collection convex, but it does not contribute photometrically.

If you use the  $-s$  option, the list of facets and normals is put to the standard output – it is useful when using a pipeline. Do not use  $-s$  and  $-v$  at the same time.

### Output parameters (out\_par)

This file contains the solution for the spin vector direction, period, and scattering parameters in the format:

```
λ β P
t0 φ0
a d k
c
```

If some of the parameters are fixed ( $t_0$  and  $\varphi_0$  are always fixed), they have the same values as in the input par file.

### Creating 3D shape (minkowski)

The shape model is fully described by its Gaussian image (**out areas**) that is used as an input for the **minkowski** procedure that creates convex polyhedron. It reads the standard input and puts the result to the standard output. The format of the output polyhedral model is as follows: the first line gives the number of vertices and facets, then follow the vertex  $x, y, z$  coordinates, then for each facet the number of vertices and the order numbers of facet vertices (anticlockwise seen from outside the body). The Minkowski problem is solved iteratively, so it takes some time (minutes).

### Triangulation (standardtri)

For some purposes it is useful to convert the polyhedron from **minkowski** to polyhedron that has all facets triangular. The **standardtri** reads the standard input (the output of **minkowski**), creates polyhedron with triangular facets and puts it to the standard output. The format is the same as for **minkowski**, but the number of vertices for each facet is omitted it is equal to three for all facets. You can use all three programs at once:

```
cat lcs | convexinv -s input_par out_lcs | minkowski | standardtri > model
```

### Finding the best period (period\_scan)

The most important step in inversion is to find the correct period. The smallest separation  $\Delta P$  of local minima in the period parameter space is roughly given by

$$\Delta P \approx 0.5 \frac{P^2}{\Delta t}$$

where  $\Delta t$  is the full epoch range of the data set. This derives from the fact that the maxima and minima of a double-sinusoidal lightcurve for periods  $P$  and  $P \pm \Delta P$  are at the same epochs after  $\Delta t$  time.

The **convexinv** program starts from initial values, finds the local minimum of  $\chi^2$  and gives the corresponding shape and spin solution. To be sure that you didn't miss the global minimum in period, it is *always absolutely necessary* to scan the whole interval of possible periods (there is usually some guess based on the synodic period and its error) before running **convexinv**.

To do this, run period scan

```
cat lcs | period_scan [-v] input_period_scan out_periods
```

The first line of the input period scan gives the initial period, the final period, and the coefficient  $p$  of the period step. The interval is scanned with the step  $p\Delta P$  (always set  $p < 1$ , recommended value is  $p = 0.8$ ). Other input parameters have the same meaning as in the **input\_par** file. The last line gives the minimum number of iterations if the iteration stop condition is smaller than one.

The procedure starts with six initial poles for each trial period, selects the period that gives the lowest  $\chi^2$  and puts it into the `out_periods` file. This file contains the period solution, rms deviation,  $\chi^2$ , the number of iteration steps, and the dark facet area (%). The number  $N_{\text{per}}$  of trial periods within the period range  $[P_0, P_1]$  is

$$N_{\text{per}} = \frac{2\Delta t(P_1 - P_0)}{P_0 P_1} \frac{1}{p}$$

so it takes some time (up to several hours for a long interval) to scan over the whole interval  $[P_0, P_1]$ .

If there is a clear minimum in  $\chi^2$  when plotted as a function of period, use this best period as the input value for `convexinv`. Run `convexinv` with different initial poles (5-10) randomly distributed over the sphere. If there is only one pole solution that gives significantly lower  $\chi^2$  than all others, you have a unique solution. There are always two possible poles with the same  $\beta$  and  $\lambda \pm 180^\circ$  for asteroids orbiting near the plane of the ecliptic.

It may happen that there is no clear minimum  $\chi^2$  versus period plot or that there are many poles giving the same residual. This means that there is not enough data for a unique period, pole, and shape model.

## Test

To run the test type:

```
cat test_lcs_rel convexinv -v -o out_areas -out_par input_convexinv out_lcs
```

and

```
cat test_lcs_rel | period_scan -v input_period_scan out_periods
```

There are lightcurves of asteroid 43 Ariadne in `Ariadne_REL_LC.TXT` (all lightcurves are relative). The best solution giving rms residual 0.016 is  $\lambda = 253^\circ$ ,  $\beta = -15^\circ$ , and  $P = 5.761985$  hours.

## Delphi Code

The Delphi code was written by Brian D. Warner, using the Durech C code for reference, except for the the Minkowski and StandardTriangles units. In order to keep stack size requirements to the default values for the Delphi compiler, large arrays were declared as objects with a single field that defined an array. For example:

```
TMaxNFACDArray = class
public
    Data : array [0..MAX_N_FAC] of Double;
end;
```

Tests against programs using the compiled C code, Delphi using a larger stack and static arrays, and Delphi using arrays as objects showed the performance to be very similar on a 2.8GHz Celeron PC. Your results may vary.

The following files comprise the Delphi core library:

uAreaNorm.pas	uEllipticalFit.pas	uMrqCof.pas
uBLMatrix.pas	uEphemerisData.pas	uMrqMin.pas
uBright.pas	uGauss.pas	uPeriodScan
uCenterOfMass.pas	uGlobals.pas	uPhaseC.pas
uConvexInv.pas	uLightcurves.pas	uSphereFunction.pas
uConvexity.pas	uLubKsb.pas	uStandardTriangles.pas
uCovSrt.pas	uLudCmp.pas	uTriangleFacets.pas
uCurvature.pas	uMatrix.pas	
uDotProduct.pas	uMinkowski.pas	

↪ The Delphi code does include conversions of the Minkowski and StandardTri FORTRAN files.

The Delphi code includes some additional units beyond those converted from the Durech C code:

### **uEphemerisData.pas**

The modeling code requires, in addition to JD/Mag (or intensity) pairs, the asteroidcentric rectangular ecliptic coordinates for the Earth and Sun. This unit provides the necessary functions or procedures to derive those values.

The ReadOrbitalElements function in this unit is empty requires the programmer provide the means for obtaining the orbital elements of an asteroid, either by MPC assigned number or name/designation.

### **uLightcurves**

This unit implements a class that manages a list of lightcurves that form a single lightcurve set. The set is then passed to the appropriate units, usually uConvexInv, for processing. Instead of objects forming the base container, this unit makes use of pointers to records, an element of which can be a TList object.

### **Delphi Output Files**

The output files using the Delphi code are nearly identical to those of the C code. Running the modeling process generates:

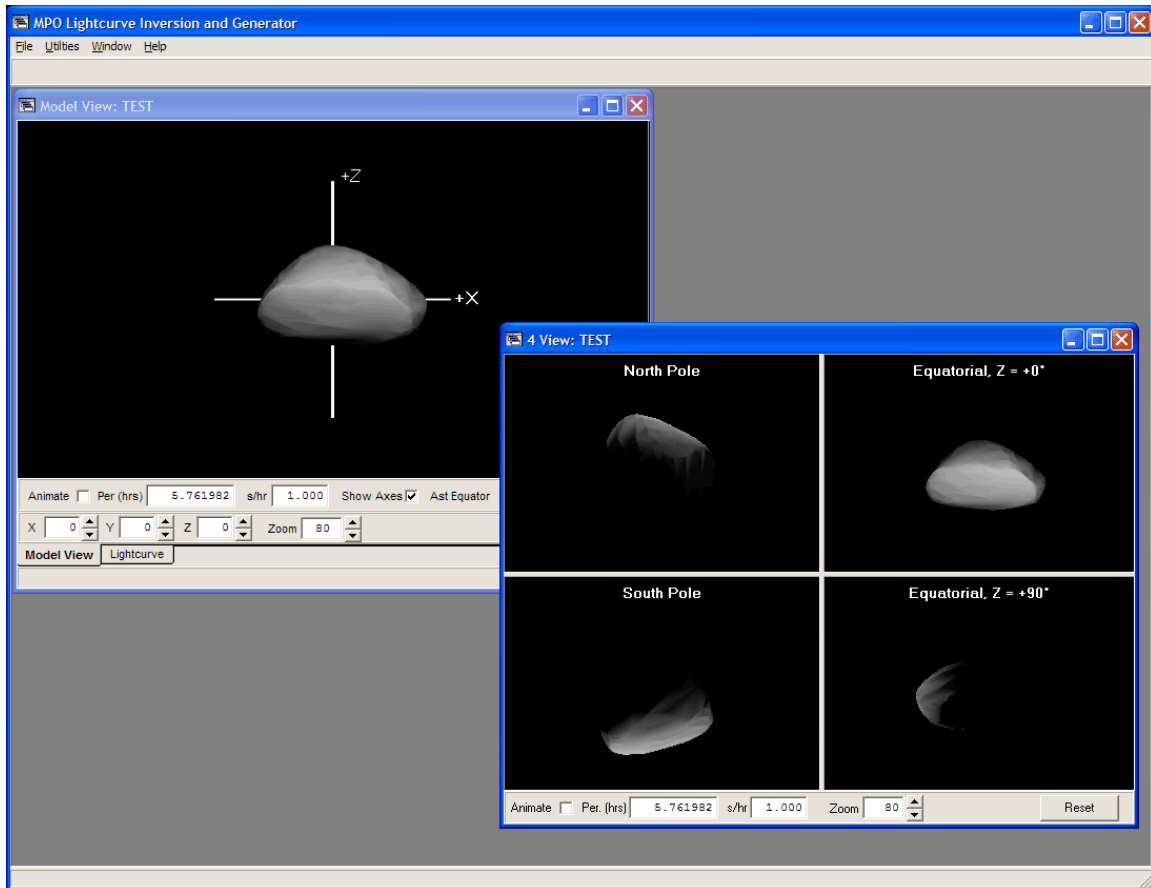
```
<BASENAME>_OUT_AREAS.TXT  
<BASENAME>_OUT_PAR.TXT  
<BASENAME>_OUT_LCS.TXT  
<BASENAME>_MINKOWSKI_SHAPE.TXT  
<BASENAME>_STDTRI.TXT
```

Where <BASENAME> is a name given by the user for the particular set.

↪ There is a significant difference between the C and Delphi version of the OUT\_LCS file. Tutorial II – AutoSearch for more details.



# LCInvert Overview



The LCInvert is a Windows-based program that uses the Delphi Inversion Library to generate the files required to display 3-D asteroid models, and then presents those models. The models can be displayed in one of two windows: Model View, which allows "orbiting" the asteroid as well as displaying it in asteroid-equatorial or ecliptic modes, and 4-Vane, which displays the asteroid from four different view points simultaneously. The views within the 4-Vane are fixed.

The 3-D presentation and animation is provided by components that use OpenGL. See "OpenGL Requirements" on page 7 for more information. The program has been tested on numerous machines and, providing the video card drivers have been updated to use OpenGL properly, it should run with no difficulties.

The main LCInvert form is what's called an "MDI parent," meaning that the Model and 4-Vane windows are "child" windows. The MDI structure allows multiple views to be open and managed by the parent.

Some of the inversion methods are very CPU intensive and so can take some time to run. During some of the processing, despite frequent calls to allow Windows to process messages, the LCInvert program can take control of a large amount of CPU time and so make other programs run slow. Future versions of the Inversion Library may incorporate so-called "worker threads" so that the processing can be put more into the background and so allow other programs to run better. The tradeoff, of course, is that the processing within the thread will be slower in order to give more CPU time to other programs.

## Main Menu

The main menu provides access to the LCInvert features. Details on the menu items are covered in later sections of this manual.

## File Menu

### Load

Displays a form that loads model data. The model data can then be selected to display a 3-D rendering of the model data.

### Clear

Clears all model data from the LCInvert program.

### Open

Opens either a Model View or 4 View window of a selected model data.

### Exit

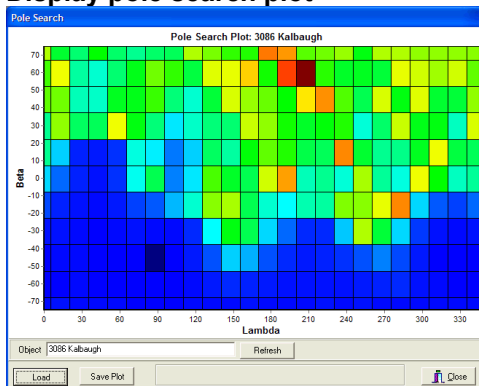
Closes the LCInvert program.

## Utilities Menu

### Invert Lightcurves

Displays the Inversion Wizard, which converts lightcurve data into model data.

### Display pole search plot



Displays a plot of the pole search results as a graphic using gradations between blue (best) and red (worst) solutions. The "ChiSq Results" file is used to create the plot. This is covered in one of the on-line tutorials.

### Standard triangles

Covers "Minkowski" model data, with polygonal facets, to triangular facets. The LCInvert 3-D engine accepts only triangular facet model data ("mesh data").

### Fix Center of Mass

Assures that the model data is centered about the center of mass of the model.

### Compute Volume

Computes the volume of the model and center of mass.

### MPCORB Convert

Converts the latest MPCORB file from the Minor Planet Center into a format used by MPO programs.

### **Import AstDys Data**

Displays the AstDys Data Import form (see Tutorial VI). This allows importing "sparse" data sets to be used for modeling.

## **Windows Menu**

### **Animate all**

Simultaneously starts or stops animation on all open Model and 4-Vane windows.

### **Close**

Closes the active Model or 4-Vane window.

### **Close all**

Closes all Model and 4-Vane windows.

## **About the Tutorials**

Several tutorials are found later in this manual. The first walks through the basic use of the Inversion Wizard using the sample data and when an approximate pole solution is known. Additional tutorials cover using the sample lightcurve data to do a search for pole solutions and importing lightcurve data from MPO Canopus.

This approach, learning by example instead of going through the Wizard form and covering every possibility without any examples, should allow you to learn using the Wizard more quickly and easily.



If path names in the tutorials don't match yours exactly, *don't panic!* The installation program can change over time and so files may be in slightly different locations under the main directory of LCINVERT. Get to that base directory and you'll find the sample files eventually. Of course, if you change the output path from that in the tutorial, then you'll need to remember that revised path if you need the files at a later time.

While the tutorials try to give you all the details, some may vary a little from your runs. The important thing is to learn the elements being presented and not get bogged down in minor details that do not affect the process or its outcome.



## Working with Model Data

---

Naturally, you want to see spinning asteroids as soon as possible. This requires that you have the right model data. To have that, you must invert a number of lightcurves. What if you don't have a number of lightcurves to invert? Fortunately, the LCInvert program comes with a sample set of model data that allows you to "spin up" quickly. It also provides the means to discuss the concepts of model data and how to work with it to display 3-D models before worrying about how that model data is generated.

### What is Model Data?

The model data tells the program what it needs to know to display a 3-D rendering of an asteroid with the proper orientation and rotation rate. Therefore, the model data for the LCInvert program consists of at least:

Sidereal Period	This value is expressed in hours
Ecliptic Longitude	The longitude of the "north pole" in ecliptic coordinates. The north pole is the one such that an observer above that pole would see the asteroid rotating anti-clockwise, just as a person above the north pole of the Earth would see the Earth rotating anticlockwise.
Ecliptic latitude	The latitude of the "north pole" in ecliptic coordinates.
Epoch	The Julian Date for a rotation angle of 0.
Rotation Angle	The initial rotation angle of the asteroid at Epoch
"Mesh Data"	This is the set of <i>triangular</i> facets and the vertices of each triangle that described a closed convex shape that is the 3-D representation of the asteroid. The process described by Kaasalainen and Torppa (2001) initially produces a set of polygonal facets, i.e., those with three or more sides. The 3-D modeling engine of LCInvert uses only triangular facets. The final step of the model processing with the Inversion method (and the Standard Triangles utility) is to break down the polygonal facets into a, usually, larger set of triangles.

There are sources for model data on the Web. Some of the available model data sets work quite well with LCInvert, others do not. See the discussion under "Shape and Spin Axis Modeling Primer" starting on page 13 for more information. In general, you want to be sure that the data were created from relative lightcurves and not absolute. As discussed in the section just mentioned, even absolute data can be considered relative under some circumstances.

The following sections will step through using the sample data. Of course, when using your own data, change the names of the files in the following steps as required.

### Relative Sizes of Models

In most cases, you'll be working with relative data (see "Shape and Spin Axis Modeling Primer"). Among other things, this means the generated models will be scaled within themselves and not on an absolute scale. Therefore, when you load models of more than one asteroid, their relative sizes among each other do not necessarily reflect the differences in size of the asteroids.

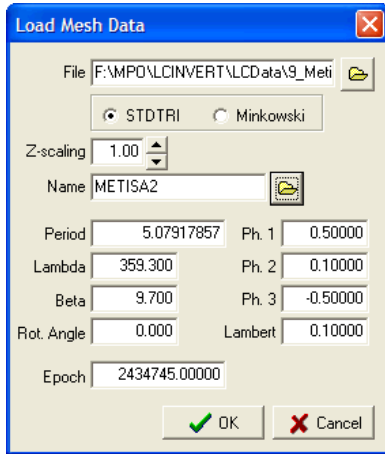
If the actual diameters are not known, you can use the following formula to derive them.

$$D(\text{km}) = 1329 * 10^{-0.2H} / \sqrt{p_v}$$

Where H is the absolute magnitude of the asteroid and  $p_v$  is the albedo.

The "zoom" setting (see below) is not directly proportional to the size of asteroids, i.e., doubling the zoom value does not have the effect of doubling the size of the asteroid for side-by-side comparisons.

## Loading Model Data



To load model data, select File | Load from the main menu. This displays the "Load Mesh Data" form.

### File

This is a read-only field, so you must click the button to its right to display a file open form. The sample data are located in the \LCINVERT\DATA directory. Select ARIADNE\_REL\_STDTRI.TXT. The "REL" indicates the data were the result of using a set of *relative* lightcurves.

### STDTRI / Minkowski

Click the radio button that correctly describes the facet (mesh) data. If the data are sets of triangles and their vertices, select STDTRI. If the data are sets of polygonal facets and their vertices, click Minkowski.

↳ If loading Minkowski data, they are converted to triangular facets before being loaded into the rendering engine.

### Z-scaling

Enter a value between 0.01 and 2.00 to scale the Z-axis (c in abc ellipse) from its original value.

Using relative data means that the Z-axis cannot be firmly established. This could result in good model being rejected because it appears to be "too tall." Use this setting to scale the model so that its Z-axis (c dimension) is no longer the longest dimension.

↳ Scaling cannot be done arbitrarily. The model lightcurve after scaling must still fit the original data. If it does not, then the model is highly suspect.

### Name

This field is filled in automatically using the base name of the data file. You can change this if you want.

### Period

Enter the sidereal period found as part of the modeling process.

↳ To load this and the remaining fields automatically, click the button next to the Period field. This displays a file open form. Locate the ARIADNE\_REL\_OUT\_PAR.TXT file associated with the mesh data and then click Open on the form. The OUT\_PAR file is discussed in detail in "The Inversion Wizard Tutorial" starting on page 35.

**Lambda**

Enter the ecliptic longitude of the North Pole of the asteroid, in degrees. The range of values is 0° to 359.9°

**Beta**

Enter the ecliptic latitude of the North Pole of the asteroid, in degrees. The range of values is -90° to +90°.

**Rot Angle**

Enter the rotation angle of the asteroid about the Z-axis, in degrees, for the given Epoch. This is usually 0°. The range of values is 0° to 359.9°.

**Epoch**

Enter the Julian Date for 0° rotation angle. This is usually the integer value of the earliest JD in the light-curve data.

**OK**

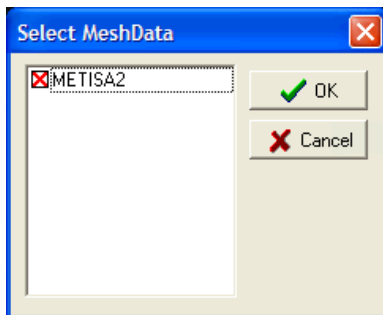
Click this button to load the data into the rendering engine.

**Cancel**

Click this button to close the form without loading data into the rendering engine.

## Opening A View

Select File | Open from the main menu. You have two options: Model view and 4-Vane. Select one of the options to display the Select Mesh Data form.

**Models List**

The list displays all the mesh data (model) sets that have been loaded, either manually through File | Load from the main menu or as a result of running the Inversion Wizard and loading one or more results.

Click on the model you want to display. This checks the box next to the name.



You can open one and only one view at a time.

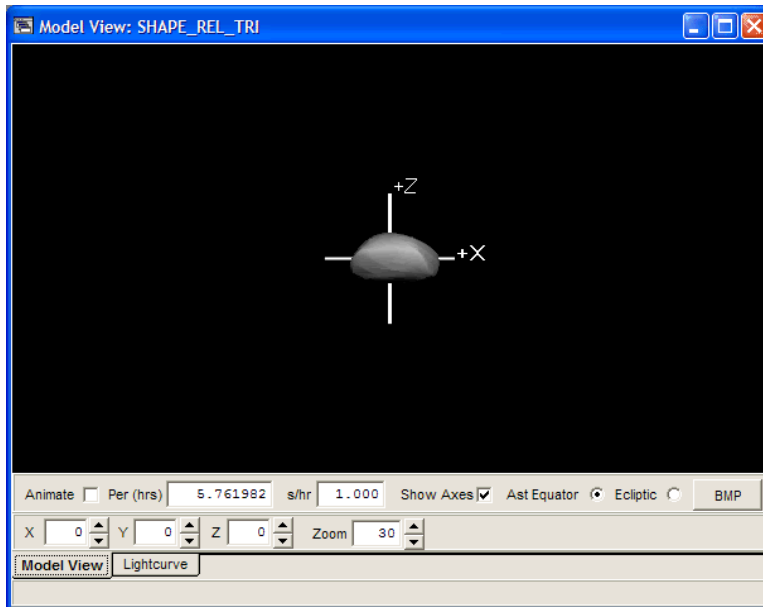
**OK**

Click this button to open the model in the selected view window.

**Cancel**

Click this button to close the form without opening a view window.

## The Model View



The Model View displays a 3-D rendering of the model that can be animated. The viewing angles and "camera zoom" can be changed to provide different looks at the model. As a check of the quality of the model, the original lightcurve data can be compared to the lightcurve data generated by the model.

## The Model View Tab

### **Animate**

Check this box to rotate the model about the Z-axis, which is the axis joining the north and south poles. Uncheck the box to stop the animation.



If more than one view is open, you can use Window | Animate all to start the animation for all windows simultaneously. If the models are for different asteroids with different rotation rates, this allows a visual demonstration of the differences in rotation rates among asteroids.

### **Per (hrs)**

Enter the sidereal period in hours. This value is normally entered automatically since it is part of the basic model data set.

### **s/hr**

Enter the rotation rate of the mode in seconds per hour. The default is 1 second = 1 hour. When displaying models for different asteroids, keep this value the same for each view so that a direct comparison can be made of the different rotation rates.

### **Show Axes**

Check this box to show the X/Y/Z axes. Labels indicate the positive direction. In the case of the Z-axis, the positive direction represents the north pole of the asteroid.

### **Ast Equator**

Check this radio button to display the model in Asteroid Equator mode. This is when the observer is directly over the equator of the asteroid, at local noon, and 0° rotation. The "sun" would be directly behind the observer.



**Ecliptic**

Check this radio button to display the model in Ecliptic mode. This places the observer on the plane of the ecliptic looking directly at the center of the asteroid. The asteroid is at either ascending or descending node, so that the observer is not "peeking" over the asteroid. The sun is directly behind the observer, on the plane of the ecliptic.

**Save**

Click this button to display a file save dialog. You can save the plot as a Windows bitmap (BMP) or PNG (compressed, nearly lossless graphics file).

**BMP**

Click this button to save a Windows bitmap of the model as currently displayed, including axes if visible.

**X**

Enter a value from 0 to 359. This sets the rotation of the X-axis to that value. During animation, this value remains constant.

**Y**

Enter a value from 0 to 359. This sets the rotation of the Y-axis to that value. During animation, this value remains constant.

**Z**

Enter a value from 0 to 359. This sets the rotation of the Y-axis to that value. During animation, this value cycles through 0 to 359 to 0.

**Zoom**

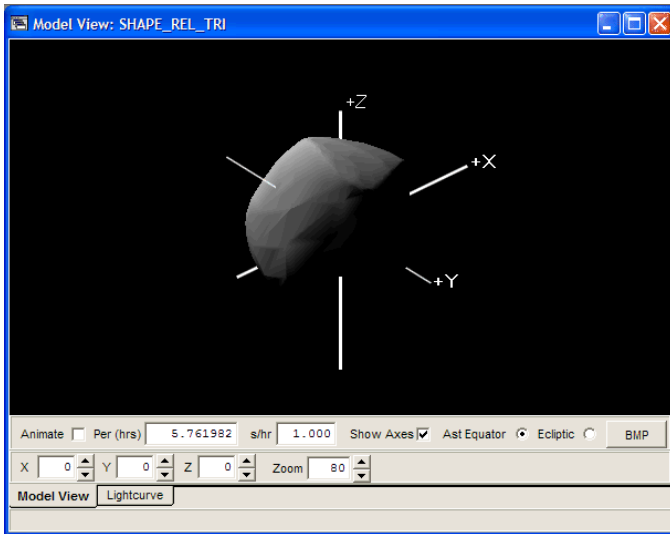
Enter a value from 1 to 250. This sets the "focal length" of the "camera" viewing the scene. Larger values zoom in (make the asteroid larger) while smaller values zoom out (make the asteroid smaller). When viewing models of different asteroids, using the same value for this field allows a comparison of the different sizes.

**Orbiting the Asteroid**

You can reposition the "camera" viewing the scene and so take a tour around the asteroid. The position of the sun does not change with the camera motion and so you can literally go to the "dark side." You can also animate the asteroid once the camera has been repositioned.

**To Orbit the Asteroid**

Depress the left mouse button while over the model display and then move the mouse (drag). As you move the mouse the "camera" changes position. Release the button to keep the camera in the current position. If the axes are being displayed, they will also rotate.

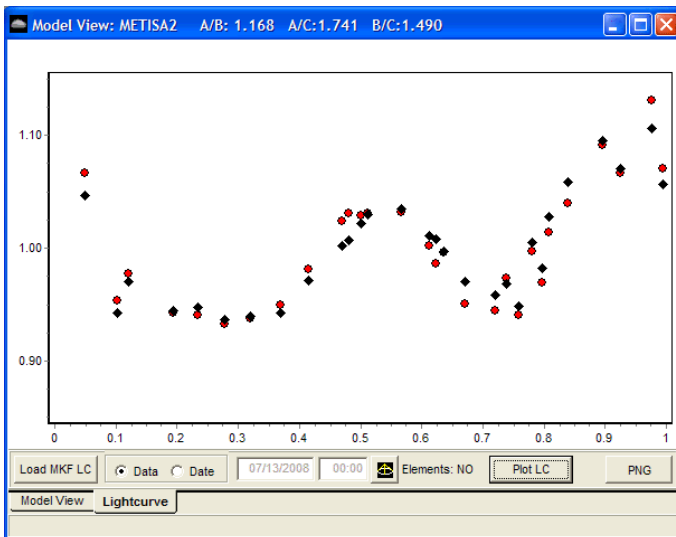


The screen shot above shows the Ast Equator view after the zoom has been set to 80 and the camera orbited to a different location.

### To Reset the View

Click the opposite view mode radio button, e.g., if the current mode is Ast Equator, click the Ecliptic button to reset the view to the default for Ecliptic view. To return to the Asteroid Equator mode, click the Ast Equator radio button. The Reset does not change the focal length of the camera.

### The Lightcurve Tab

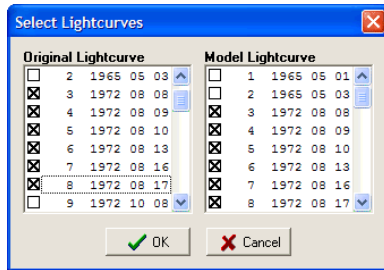


The Lightcurve tab of the Model View allows you to compare the original lightcurve data against the model's lightcurve. If the two match well, then you have some added confidence in the model solution.

The lightcurve is based on the loaded model data, i.e., you can generate curves for only those dates covered by the original lightcurves used to generate the model. Such an example is in the screen shot above.

1. Click the Load MKF LC button; this displays a file open form. Locate and load the Kaasalainen-format file that was used to find the model. This step is required only if the file has not been previously loaded while the model view form was open.
2. Click the Data radio button.

3. Click the Plot LC button. This displays the Select Lightcurves form.



The form has two lists. Both display the dates for the individual lightcurves that formed the total lightcurve file. The two lists should be identical since the Model lightcurves are simply conversions of the Original lightcurve.

### Selecting Lightcurve Sets

To select a set for comparison, check (click on) one or more of the boxes in the Original lightcurve list. This automatically checks the corresponding boxes in the Model list. Both lists are multi-select, meaning they follow the usual behavior of Windows list controls.

1. To select a range of items, click on the first item and then Shift-click on the last item.
2. To select multiple but non-contiguous items, Ctrl+click on each item.
3. Clicking without Shift or Ctrl on any item clears all entries and selects only the one that was clicked.



Do not select every available lightcurve. Instead, select only sets of lightcurves that were made about the same time. This keeps the phase angle of the asteroid about the same and, therefore, the shape of the curve is about the same.

### OK

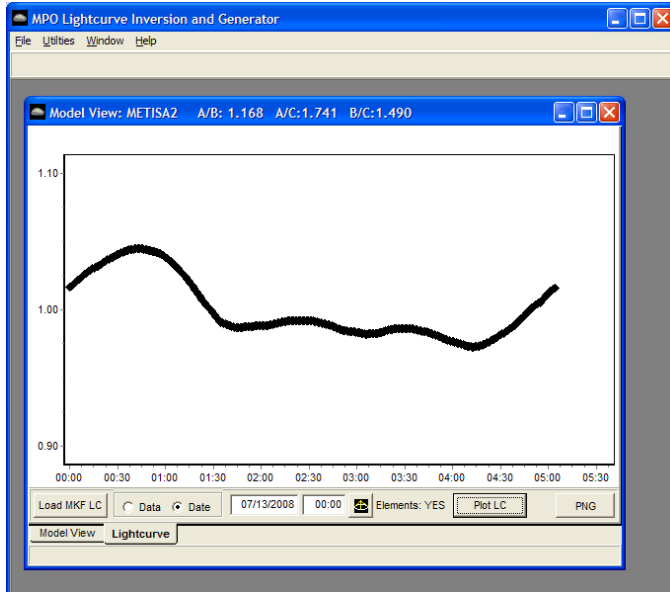
Click this button to display the selected Original and Model lightcurves. Red circles represent the Original lightcurve data while black diamonds represent the Model lightcurve data.

### Cancel

Click this button close the selection form without plotting any data.

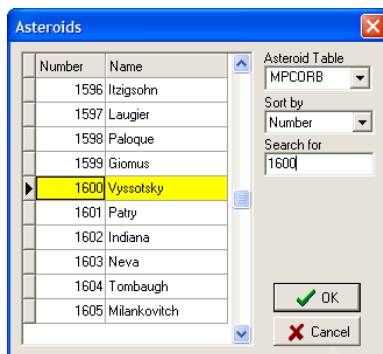
4. Click this button to save the current lightcurve plot comparison.

## Arbitrary Date



The lightcurve shows the model curve for the select date. This allows you to compare observations on that date against the model curve or to predict the lightcurve for planning purposes.

1. Click the Load MKF LC button; this displays a file open form. Locate and load the Kaasalainen-format file that was used to find the model. This step is required only if the file has not been previously loaded while the model view form was open.
2. Click the Date radio button.
3. Enter the date and time (UT) for the start of the lightcurve. Use mm/dd/yyyy format with leading zeros for the date and hh:mm, 24-hour format, for the time, also using leading zeros if required.
4. If not previously done, click the speed button to the right of the time field. This displays a form to load the elements for the asteroid.



### Asteroid Table

Select the table from which the elements are taken

MPCORB (Minor Planet Center)

ASTORB (Lowell Observatory)

USER (User-entered elements)



If LCinvert is the only MPO software you use, then you can update only the MPCORB file and so you should always select this table. If you use other MPO software (e.g., MPO

Canopus) and update the ASTORB and/or USERELMS files, then those tables can be used as well.

### Sort by

Select the sort order from the drop down list: Number or Name

### Search for

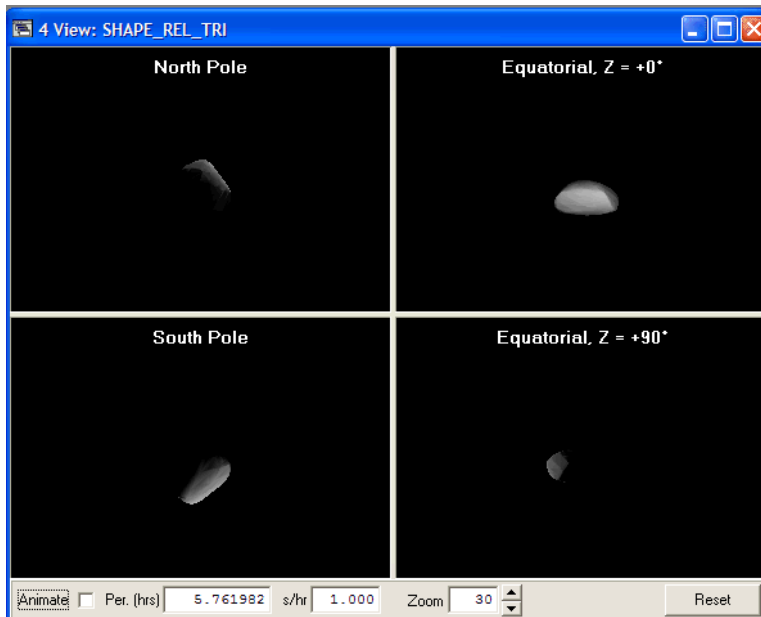
Enter all or part of an entry appropriate to sort order. Pause a moment and the table jumps to the nearest match to the entry in this field.

5. Click the Plot LC button to generate the lightcurve.

↵ The Plot LC button is disabled if the Date radio button is selected and no elements are loaded.

6. Click the Save button to save the plot.

## The 4-Vane View



The 4-Vane View displays four fixed views of the selected model: from over the north pole, from over the south pole, from above the equator at local noon, and from the equator at the sunrise line.

### Animate

Click this button to animate all four panes simultaneously. If more than one view is open, Model and/or 4-Vane, all views can be animated simultaneously by selecting Window | Animate all from the main menu.

### Per (hrs)

Enter the sidereal period in hours. This value is normally entered automatically since it is part of the basic model data set.

### s/hr

Enter the rotation rate of the mode in seconds per hour. The default is 1 second = 1 hour. When displaying models for different asteroids, keep this value the same for each view so that a direct comparison can be made of the different rotation rates.

**Zoom**

Enter a value from 1 to 250. This sets the "focal length" of the "camera" viewing the scene. Larger values zoom in (make the asteroid larger) while smaller values zoom out (make the asteroid smaller). When viewing models of different asteroids, using the same value for this field allows a comparison of the different sizes.

**Reset**

Click this button to reset the Z-axis rotation to 0°. The focal length of the camera is not reset.

## Tutorials

---

The on-line help file contains a number of tutorials that step you through the use of LCInvert. Given the many possibilities, it was felt that a set of tutorials, each one going through the steps for a common approach to modeling in LCInvert, was better than trying to describe things “in general” and hope the user could figure things out.

Most of the tutorials use data that were installed when you installed LCInvert. Look in the \LCINVERT\DATA directory.



It's a good idea to make a backup of all the sample data files before you start working with LCInvert. That way, you can restore them to their original state should you accidentally modify or delete them while working through the tutorials.



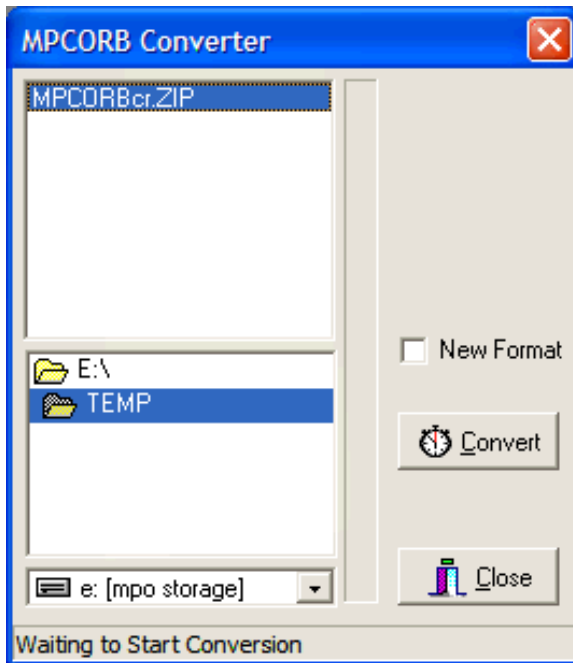


## Data Conversions

LCInvert has two important data conversion utilities. One updates the MPCORB file from the Minor Planet Center and so gives you the latest orbital elements. These are important for computing phase corrections and the asteroid centric positions of the Sun and Earth for modeling.

The other allows you to import observations from the AstDys site, which gathers the data from the MPC reports and other sources. These give you the “sparse” data needed to do modeling. One of the tutorials in the on-line help covers in detail downloading and converting that data.

### MPCORB Convert



This program converts the MPCORB database. This program alters MPCORB.FF2.

Select Utilities | MPCOrb Convert from the main menu to display the form.

#### File

Use this to find MPRCORB\*.ZIP or MPCORB\*.DAT file. If you downloaded the ZIP file, the program automatically unzips it for you before converting.

#### New Format

Check this box to convert the data to the expanded MPC format. ***Do NOT check this box unless and until the MPC adopts the new format.***

#### Convert

Click to start the conversion process

#### Close

Click to close the program. This does not cancel or reverse the conversion process if it was run.

## **Steps to Convert the Files**

1. Download MPCORB\*.ZIP or MPCORB\*.DAT file.
2. Locate MPCORB\*.ZIP or MPCORB\*.DAT file in the directory/file controls
3. Click the Convert button.